

bn00aa

Computes $\mathbf{B}_P \cdot \mathbf{e}_\theta \times \mathbf{e}_\zeta$ on computational boundary, $\partial\mathcal{D}$.

[called by: [xspech](#).]

[calls: [coords](#) and [casing](#).]

contents

| | | |
|----------|--|----------|
| 1 | bn00aa | 1 |
| 1.1 | free-boundary constraint | 1 |
| 1.2 | construction of normal field | 1 |
| 1.3 | outline | 1 |

1.1 free-boundary constraint

1. The normal field at the computational boundary, $\partial\mathcal{D}$, should be equal to $(\mathbf{B}_P + \mathbf{B}_C) \cdot \mathbf{n}$, where \mathbf{B}_P is the “plasma” field, i.e. the magnetic field produced by internal plasma currents, and is computed using virtual casing, and \mathbf{B}_C is the “vacuum” field, i.e. the magnetic field produced by the external coils, and is given on input.

1.2 construction of normal field

1. The normal depends on geometry:

Igeometry.eq.1 : Cartesian ;

Igeometry.eq.2 : Cylindrical ;

Igeometry.eq.3 : Toroidal ;

$$\mathbf{e}_\theta \times \mathbf{e}_\zeta = -R Z_\theta \hat{r} + (Z_\theta R_\zeta - R_\theta Z_\zeta) \hat{\phi} + R R_\theta \hat{z}.$$

1.3 outline

1. The computational boundary is obtained using [coords](#). (Note that this does not change, so this needs only to be determined once.)
2. At each point on the computational boundary (i.e., on the discrete grid), [casing](#) is used to compute the plasma field using the virtual casing principle.
3. The transformation from cylindrical to Cartesian is

$$x = R \cos \zeta, \quad y = R \sin \zeta, \quad z = Z. \tag{1}$$

This induces the vector transformation

$$\begin{aligned} B^R &= +B_x \cos \zeta + B_y \sin \zeta \\ B^\phi &= (-B_x \sin \zeta + B_y \cos \zeta)/R \\ B^Z &= B_z \end{aligned} \tag{2}$$